

XMLLib

0.1

Généré par Doxygen 1.7.2

Wed Apr 20 2011 21 :30 :45



# Table des matières

<b>1</b>	<b>Index des espaces de nommage</b>	<b>1</b>
1.1	Liste des espaces de nommage . . . . .	1
<b>2</b>	<b>Index des classes</b>	<b>3</b>
2.1	Hiérarchie des classes . . . . .	3
<b>3</b>	<b>Index des classes</b>	<b>5</b>
3.1	Liste des classes . . . . .	5
<b>4</b>	<b>Index des fichiers</b>	<b>7</b>
4.1	Liste des fichiers . . . . .	7
<b>5</b>	<b>Documentation des espaces de nommage</b>	<b>9</b>
5.1	Référence de l'espace de nommage xml . . . . .	9
<b>6</b>	<b>Documentation des classes</b>	<b>11</b>
6.1	Référence de la classe xml : :CompositeMarkupNode . . . . .	11
6.1.1	Description détaillée . . . . .	14
6.1.2	Documentation des définitions de type membres . . . . .	14
6.1.2.1	_Children . . . . .	14
6.1.2.2	Children . . . . .	14
6.1.2.3	ChildrenIterator . . . . .	14
6.1.3	Documentation des constructeurs et destructeur . . . . .	15
6.1.3.1	CompositeMarkupNode . . . . .	15
6.1.3.2	~CompositeMarkupNode . . . . .	15
6.1.4	Documentation des fonctions membres . . . . .	15
6.1.4.1	accept . . . . .	15
6.1.4.2	begin . . . . .	15
6.1.4.3	end . . . . .	15
6.1.5	Documentation des données membres . . . . .	16
6.1.5.1	_children . . . . .	16
6.1.5.2	_proxyToSelf . . . . .	16
6.2	Référence de la classe xml : :DotOutputVisitor . . . . .	16
6.2.1	Description détaillée . . . . .	19
6.2.2	Documentation des constructeurs et destructeur . . . . .	19
6.2.2.1	DotOutputVisitor . . . . .	19
6.2.2.2	~DotOutputVisitor . . . . .	19
6.2.3	Documentation des fonctions membres . . . . .	20
6.2.3.1	declareNode . . . . .	20
6.2.3.2	visit . . . . .	20

6.2.3.3	visit	20
6.2.3.4	visit	20
6.2.3.5	writeAttributes	20
6.2.3.6	writeDot	20
6.2.3.7	writeld	20
6.2.4	Documentation des données membres	20
6.2.4.1	_document	20
6.2.4.2	_nodesCounter	21
6.2.4.3	_out	21
6.2.4.4	GRAPH.TYPE	21
6.2.4.5	LINK	21
6.2.4.6	textNodesCount	21
6.3	Référence de la classe xml : :InterfaceNodeVisitor	21
6.3.1	Description détaillée	23
6.3.2	Documentation des constructeurs et destructeur	23
6.3.2.1	~InterfaceNodeVisitor	23
6.3.3	Documentation des fonctions membres	23
6.3.3.1	visit	23
6.3.3.2	visit	24
6.3.3.3	visit	24
6.3.4	Documentation des fonctions amies et associées	24
6.3.4.1	CompositeMarkupNode	24
6.3.4.2	MarkupNode	24
6.3.4.3	TextNode	24
6.4	Référence de la classe xml : :MarkupNode	24
6.4.1	Description détaillée	27
6.4.2	Documentation des définitions de type membres	27
6.4.2.1	_Attributes	27
6.4.2.2	Attributes	27
6.4.2.3	AttributesIteator	27
6.4.3	Documentation des constructeurs et destructeur	28
6.4.3.1	MarkupNode	28
6.4.3.2	~MarkupNode	28
6.4.4	Documentation des fonctions membres	28
6.4.4.1	accept	28
6.4.4.2	begin	28
6.4.4.3	end	28
6.4.4.4	name	28
6.4.4.5	ns	29
6.4.5	Documentation des données membres	29
6.4.5.1	_attributes	29
6.4.5.2	_name	29
6.4.5.3	_namespace	29
6.5	Référence de la classe xml : :Node	29
6.5.1	Description détaillée	32
6.5.2	Documentation des constructeurs et destructeur	32
6.5.2.1	Node	32
6.5.2.2	~Node	32
6.5.3	Documentation des fonctions membres	32
6.5.3.1	accept	32

6.5.3.2	parent	32
6.5.4	Documentation des données membres	32
6.5.4.1	_parent	32
6.6	Référence de la classe xml : :OutputVisitor	33
6.6.1	Description détaillée	36
6.6.2	Documentation des constructeurs et destructeur	36
6.6.2.1	OutputVisitor	36
6.6.2.2	~OutputVisitor	36
6.6.3	Documentation des fonctions membres	37
6.6.3.1	doIndent	37
6.6.3.2	visit	37
6.6.3.3	visit	37
6.6.3.4	visit	37
6.6.3.5	writeMarkupOpening	37
6.6.4	Documentation des données membres	38
6.6.4.1	_indent	38
6.6.4.2	_indentChar	38
6.6.4.3	_indentUnit	38
6.6.4.4	_out	38
6.6.4.5	ASSIGN_ATTRIBUTE_VALUE_STR	38
6.6.4.6	CLOSE_ATTRIBUTE_VALUE_STR	38
6.6.4.7	CLOSE_MARKUP_STR	38
6.6.4.8	CLOSING_MARKUP_STR	38
6.6.4.9	DEFAULT_INDENT_CHAR	38
6.6.4.10	DEFAULT_INDENT_UNIT	39
6.6.4.11	INSIDE_MARKUP_SPACE_STR	39
6.6.4.12	NS_SEPARATOR_STR	39
6.6.4.13	OPEN_ATTRIBUTE_VALUE_STR	39
6.6.4.14	OPEN_MARKUP_STR	39
6.7	Référence de la classe xml : :TextNode	39
6.7.1	Description détaillée	42
6.7.2	Documentation des constructeurs et destructeur	42
6.7.2.1	TextNode	42
6.7.2.2	~TextNode	42
6.7.3	Documentation des fonctions membres	42
6.7.3.1	accept	42
6.7.3.2	content	42
6.7.4	Documentation des données membres	43
6.7.4.1	_content	43
<b>7</b>	<b>Documentation des fichiers</b>	<b>45</b>
7.1	Référence du fichier src/CompositeMarkupNode.cpp	45
7.2	Référence du fichier src/CompositeMarkupNode.hh	46
7.3	Référence du fichier src/DotOutputVisitor.cpp	48
7.4	Référence du fichier src/DotOutputVisitor.hh	48
7.5	Référence du fichier src/InterfaceNodeVisitor.hpp	50
7.6	Référence du fichier src/MarkupNode.cpp	50
7.7	Référence du fichier src/MarkupNode.hh	51
7.8	Référence du fichier src/Node.cpp	53
7.9	Référence du fichier src/Node.hh	53

7.10	Référence du fichier <code>src/OutputVisitor.cpp</code>	54
7.11	Référence du fichier <code>src/OutputVisitor.hh</code>	55
7.12	Référence du fichier <code>src/TextNode.cpp</code>	56
7.13	Référence du fichier <code>src/TextNode.hh</code>	57

# Chapitre 1

## Index des espaces de nommage

### 1.1 Liste des espaces de nommage

Liste de tous les espaces de nommage avec une brève description :

<a href="#">xml</a> . . . . .	9
-------------------------------	---





## Chapitre 2

# Index des classes

### 2.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

xml : :InterfaceNodeVisitor . . . . .	21
xml : :DotOutputVisitor . . . . .	16
xml : :OutputVisitor . . . . .	33
xml : :Node . . . . .	29
xml : :MarkupNode . . . . .	24
xml : :CompositeMarkupNode . . . . .	11
xml : :TextNode . . . . .	39



## Chapitre 3

# Index des classes

### 3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<a href="#">xml : :CompositeMarkupNode</a>	11
<a href="#">xml : :DotOutputVisitor</a>	16
<a href="#">xml : :InterfaceNodeVisitor</a>	21
<a href="#">xml : :MarkupNode</a>	24
<a href="#">xml : :Node</a>	29
<a href="#">xml : :OutputVisitor</a>	33
<a href="#">xml : :TextNode</a>	39



## Chapitre 4

# Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

src/ <a href="#">CompositeMarkupNode.cpp</a>	45
src/ <a href="#">CompositeMarkupNode.hh</a>	46
src/ <a href="#">DotOutputVisitor.cpp</a>	48
src/ <a href="#">DotOutputVisitor.hh</a>	48
src/ <a href="#">InterfaceNodeVisitor.hpp</a>	50
src/ <a href="#">MarkupNode.cpp</a>	50
src/ <a href="#">MarkupNode.hh</a>	51
src/ <a href="#">Node.cpp</a>	53
src/ <a href="#">Node.hh</a>	53
src/ <a href="#">OutputVisitor.cpp</a>	54
src/ <a href="#">OutputVisitor.hh</a>	55
src/ <a href="#">TextNode.cpp</a>	56
src/ <a href="#">TextNode.hh</a>	57



## Chapitre 5

# Documentation des espaces de nommage

### 5.1 Référence de l'espace de nommage xml

#### Classes

- class [CompositeMarkupNode](#)
- class [DotOutputVisitor](#)
- class [InterfaceNodeVisitor](#)
- class [MarkupNode](#)
- class [Node](#)
- class [OutputVisitor](#)
- class [TextNode](#)





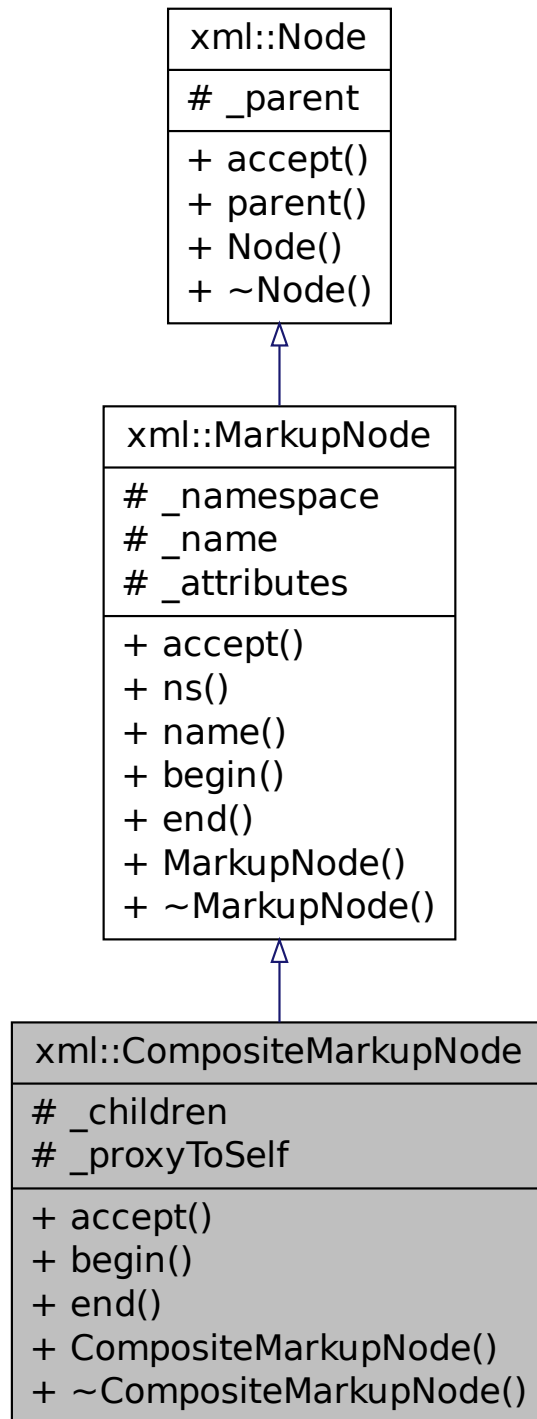
## Chapitre 6

# Documentation des classes

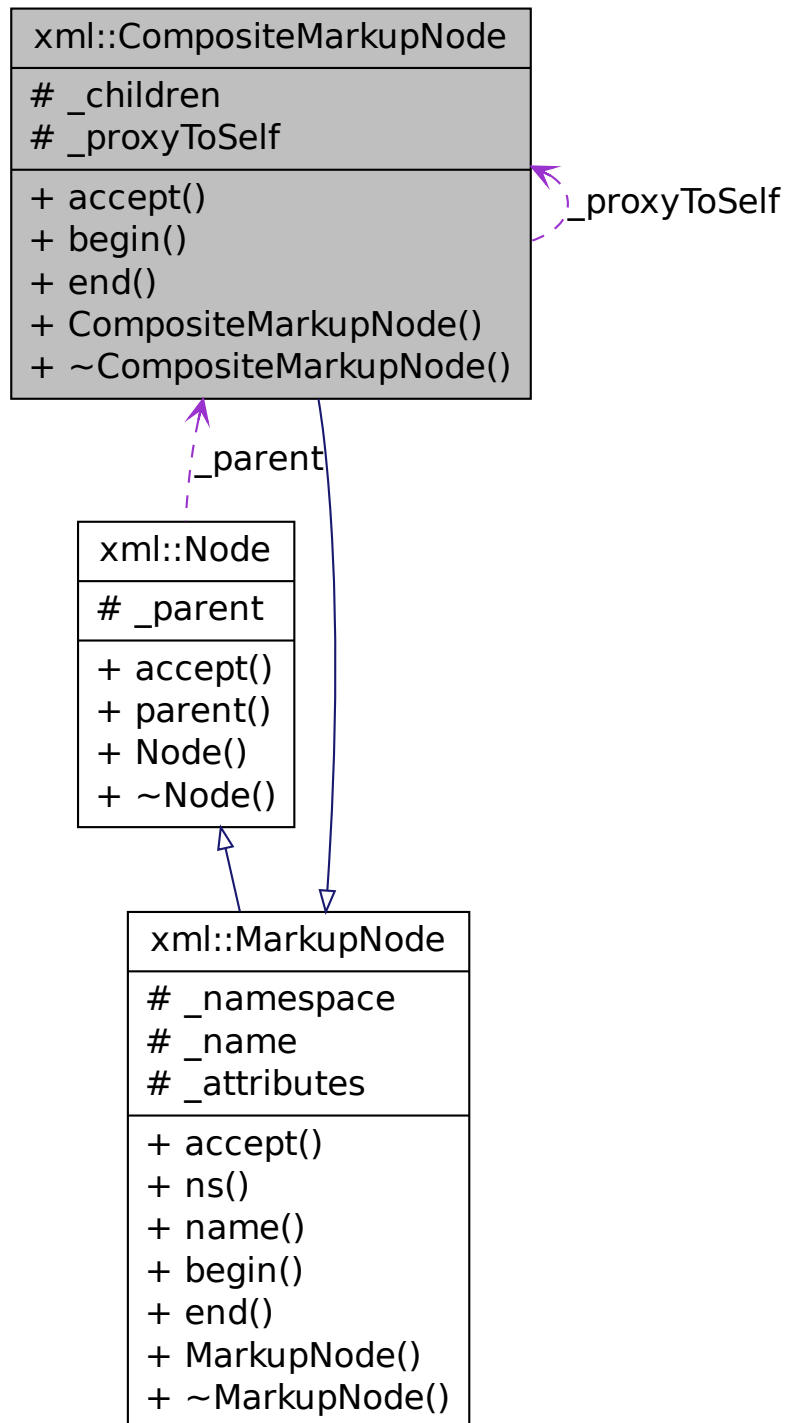
### 6.1 Référence de la classe xml : :CompositeMarkupNode

```
#include <CompositeMarkupNode.hh>
```

Graphe d'héritage de xml : :CompositeMarkupNode :



Graphe de collaboration de xml : :CompositeMarkupNode :



## Types publics

- typedef std : :list< [Node](#) \* > [Children](#)
- typedef \_Children : :const\_iterator [ChildrenIterator](#)

## Fonctions membres publiques

- virtual void [accept](#) ([InterfaceNodeVisitor](#) &visitor) const
- [ChildrenIterator](#) [begin](#) () const
- [ChildrenIterator](#) [end](#) () const
- [CompositeMarkupNode](#) ([CompositeMarkupNode](#) \*\*parent, const std : :string &ns, const std : :string &name, const [Attributes](#) &attributes, [CompositeMarkupNode](#) \*&proxyToSelf, const [Children](#) &children)
- virtual [~CompositeMarkupNode](#) ()

## Types protégés

- typedef std : :list< [Node](#) \* > [\\_Children](#)

## Attributs protégés

- [Children](#) [\\_children](#)
- [CompositeMarkupNode](#) \*& [\\_proxyToSelf](#)

### 6.1.1 Description détaillée

Définition à la ligne 19 du fichier CompositeMarkupNode.hh.

### 6.1.2 Documentation des définitions de type membres

- 6.1.2.1 typedef std : :list<[Node](#)\*> [xml](#) : :[CompositeMarkupNode](#) : :[\\_Children](#)  
[protected]

Définition à la ligne 22 du fichier CompositeMarkupNode.hh.

- 6.1.2.2 typedef std : :list<[Node](#)\*> [xml](#) : :[CompositeMarkupNode](#) : :[Children](#)

Définition à la ligne 27 du fichier CompositeMarkupNode.hh.

- 6.1.2.3 typedef [\\_Children](#) : :const\_iterator [xml](#) : :[CompositeMarkupNode](#) : :[ChildrenIterator](#)

Définition à la ligne 28 du fichier CompositeMarkupNode.hh.

### 6.1.3 Documentation des constructeurs et destructeur

6.1.3.1 `xml : :CompositeMarkupNode : :CompositeMarkupNode ( CompositeMarkupNode  
** parent, const std : :string & ns, const std : :string & name, const Attributes &  
attributes, CompositeMarkupNode *& proxyToSelf, const Children & children )`

Définition à la ligne 49 du fichier `CompositeMarkupNode.cpp`.

6.1.3.2 `xml : :CompositeMarkupNode : :~CompositeMarkupNode ( ) [virtual]`

Définition à la ligne 60 du fichier `CompositeMarkupNode.cpp`.

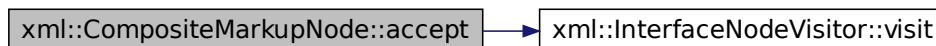
### 6.1.4 Documentation des fonctions membres

6.1.4.1 `void xml : :CompositeMarkupNode : :accept ( InterfaceNodeVisitor & visitor ) const  
[virtual]`

Réimplémentée à partir de [xml : :MarkupNode](#).

Définition à la ligne 30 du fichier `CompositeMarkupNode.cpp`.

Voici le graphe d'appel pour cette fonction :



6.1.4.2 `CompositeMarkupNode : :ChildrenIterator xml : :CompositeMarkupNode : :begin ( )  
const`

Réimplémentée à partir de [xml : :MarkupNode](#).

Définition à la ligne 35 du fichier `CompositeMarkupNode.cpp`.

6.1.4.3 `CompositeMarkupNode : :ChildrenIterator xml : :CompositeMarkupNode : :end ( ) const`

Réimplémentée à partir de [xml : :MarkupNode](#).

Définition à la ligne 40 du fichier `CompositeMarkupNode.cpp`.

## 6.1.5 Documentation des données membres

### 6.1.5.1 `_Children xml : :CompositeMarkupNode : :_children` `[protected]`

Définition à la ligne 93 du fichier `CompositeMarkupNode.hh`.

### 6.1.5.2 `CompositeMarkupNode*& xml : :CompositeMarkupNode : :_proxyToSelf` `[protected]`

Définition à la ligne 95 du fichier `CompositeMarkupNode.hh`.

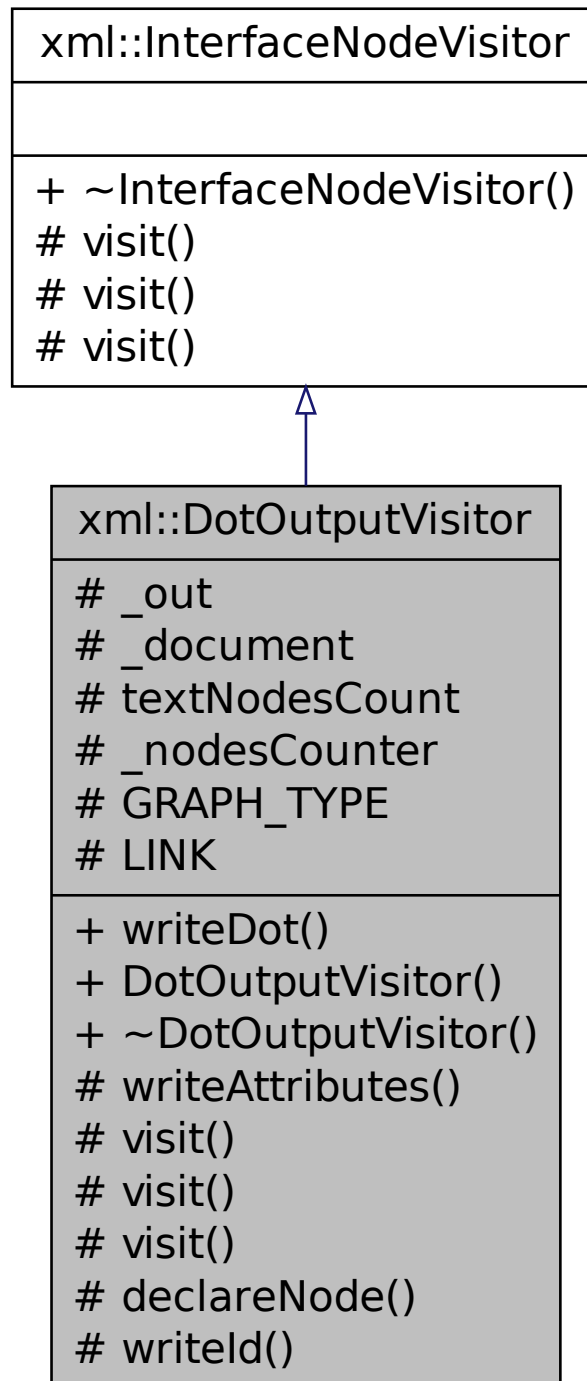
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/CompositeMarkupNode.hh](#)
- [src/CompositeMarkupNode.cpp](#)

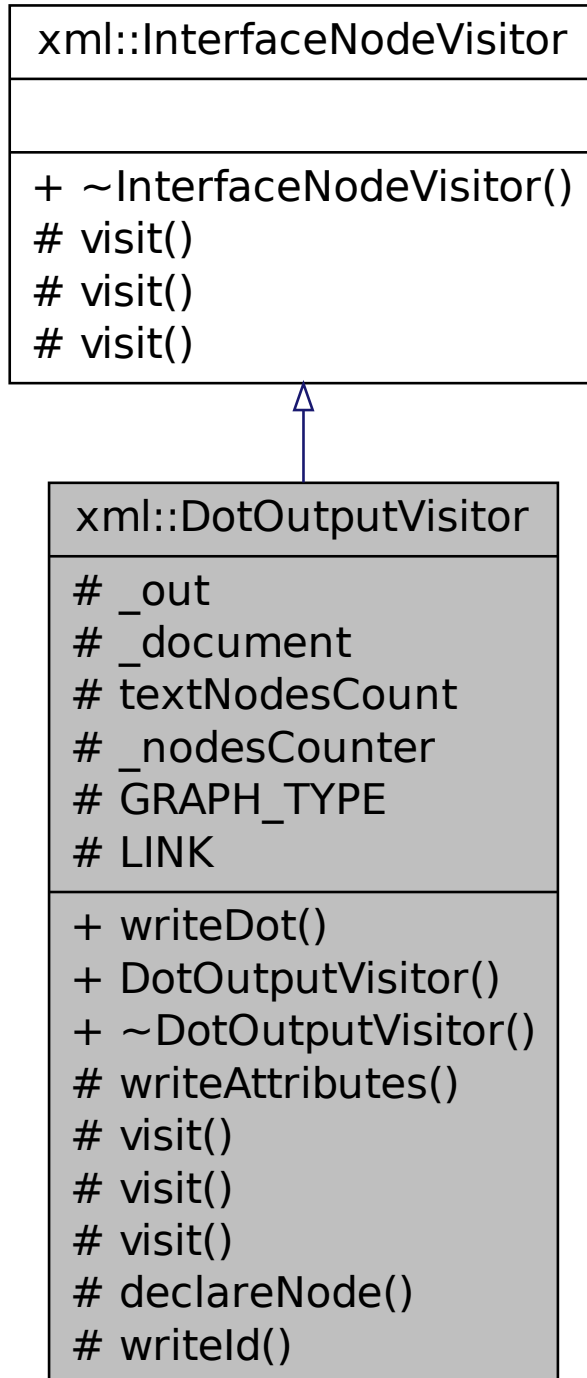
## 6.2 Référence de la classe `xml : :DotOutputVisitor`

```
#include <DotOutputVisitor.hh>
```

Graphe d'héritage de xml : :DotOutputVisitor :



Graphe de collaboration de xml : :DotOutputVisitor :





### Fonctions membres publiques

- void `writeDot` (`Node` \*node)
- `DotOutputVisitor` (`std` : :ostream &out, `std` : :string graphName)
- virtual `~DotOutputVisitor` ()

### Fonctions membres protégées

- void `writeAttributes` (const `MarkupNode` &node)
- virtual void `visit` (const `TextNode` &node)
- virtual void `visit` (const `MarkupNode` &node)
- virtual void `visit` (const `CompositeMarkupNode` &node)
- void `declareNode` (const `Node` &node, `std` : :string label)
- void `writelid` (const `Node` &node)

### Attributs protégés

- `std` : :ostream & `_out`
- `std` : :string `_document`
- int `textNodesCount`
- int `_nodesCounter`

### Attributs protégés statiques

- static const `std` : :string `GRAPH_TYPE` = "digraph"
- static const `std` : :string `LINK` = "-> "

#### 6.2.1 Description détaillée

Définition à la ligne 24 du fichier DotOutputVisitor.hh.

#### 6.2.2 Documentation des constructeurs et destructeur

6.2.2.1 `xml : :DotOutputVisitor : :DotOutputVisitor ( std : :ostream & out, std : :string graphName )`

6.2.2.2 `xml : :DotOutputVisitor : :~DotOutputVisitor ( )` [virtual]

Définition à la ligne 49 du fichier DotOutputVisitor.cpp.

### 6.2.3 Documentation des fonctions membres

**6.2.3.1** `void xml : :DotOutputVisitor : :declareNode ( const Node & node, std : :string label )`  
[protected]

**6.2.3.2** `void xml : :DotOutputVisitor : :visit ( const MarkupNode & node )` [protected,  
virtual]

Implémente [xml : :InterfaceNodeVisitor](#).

Définition à la ligne 85 du fichier DotOutputVisitor.cpp.

**6.2.3.3** `void xml : :DotOutputVisitor : :visit ( const CompositeMarkupNode & node )`  
[protected, virtual]

Implémente [xml : :InterfaceNodeVisitor](#).

Définition à la ligne 92 du fichier DotOutputVisitor.cpp.

**6.2.3.4** `void xml : :DotOutputVisitor : :visit ( const TextNode & node )` [protected,  
virtual]

Implémente [xml : :InterfaceNodeVisitor](#).

Définition à la ligne 77 du fichier DotOutputVisitor.cpp.

**6.2.3.5** `void xml : :DotOutputVisitor : :writeAttributes ( const MarkupNode & node )`  
[protected]

Définition à la ligne 59 du fichier DotOutputVisitor.cpp.

**6.2.3.6** `void xml : :DotOutputVisitor : :writeDot ( Node * node )`

Définition à la ligne 69 du fichier DotOutputVisitor.cpp.

**6.2.3.7** `void xml : :DotOutputVisitor : :writeld ( const Node & node )` [protected]

Définition à la ligne 114 du fichier DotOutputVisitor.cpp.

### 6.2.4 Documentation des données membres

**6.2.4.1** `std : :string xml : :DotOutputVisitor : :_document` [protected]

Définition à la ligne 62 du fichier DotOutputVisitor.hh.

**6.2.4.2** `int xml : :DotOutputVisitor : :_nodesCounter` `[protected]`

Définition à la ligne 64 du fichier DotOutputVisitor.hh.

**6.2.4.3** `std : :ostream& xml : :DotOutputVisitor : :_out` `[protected]`

Définition à la ligne 61 du fichier DotOutputVisitor.hh.

**6.2.4.4** `const std : :string xml : :DotOutputVisitor : :GRAPH_TYPE = "digraph"` `[static, protected]`

Définition à la ligne 58 du fichier DotOutputVisitor.hh.

**6.2.4.5** `const std : :string xml : :DotOutputVisitor : :LINK = "->"` `[static, protected]`

Définition à la ligne 59 du fichier DotOutputVisitor.hh.

**6.2.4.6** `int xml : :DotOutputVisitor : :textNodesCount` `[protected]`

Définition à la ligne 63 du fichier DotOutputVisitor.hh.

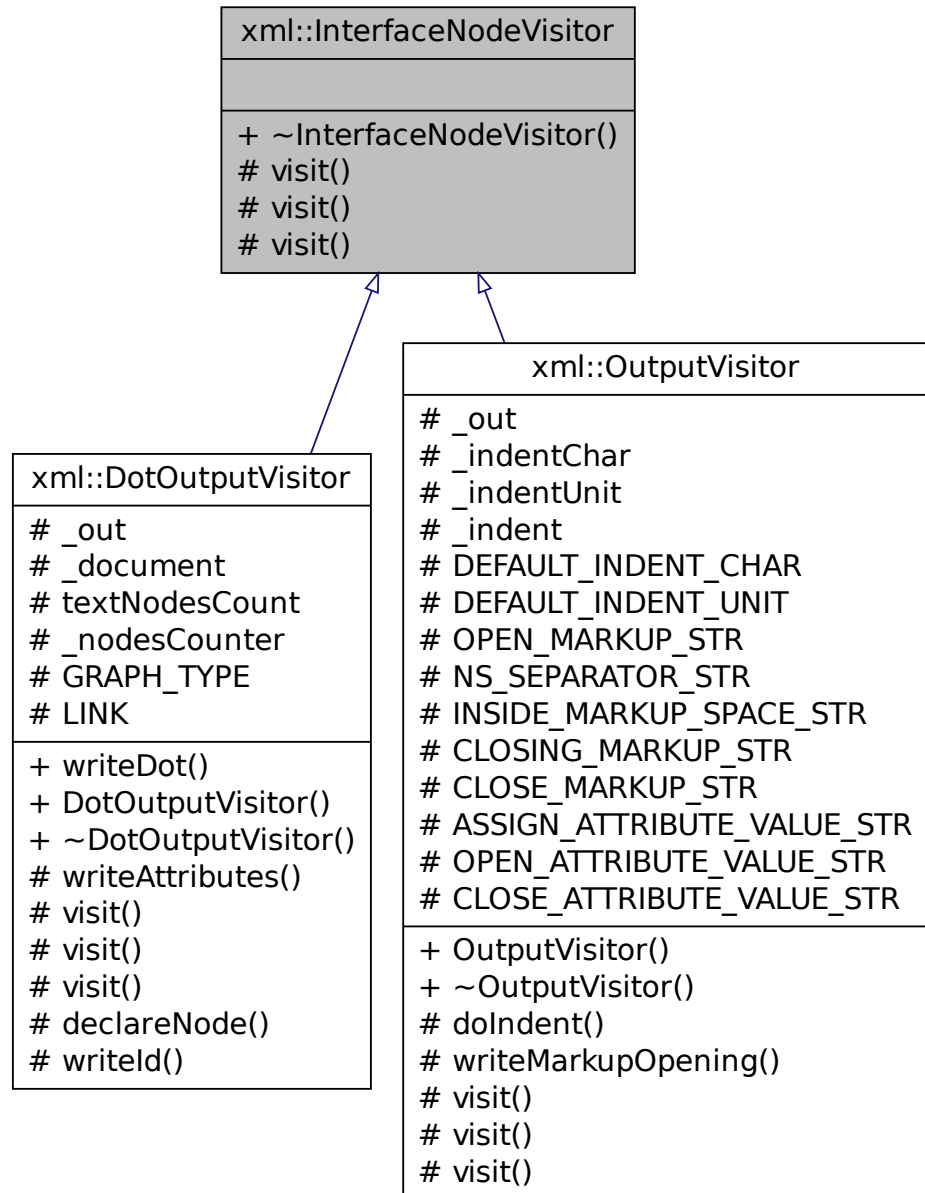
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/DotOutputVisitor.hh](#)
- [src/DotOutputVisitor.cpp](#)

## 6.3 Référence de la classe xml : :InterfaceNodeVisitor

```
#include <InterfaceNodeVisitor.hpp>
```

Graphe d'héritage de xml : :InterfaceNodeVisitor :



### Fonctions membres publiques

– virtual [~InterfaceNodeVisitor](#) ()

### Fonctions membres protégées

- virtual void [visit](#) (const [TextNode](#) &node)=0
- virtual void [visit](#) (const [MarkupNode](#) &node)=0
- virtual void [visit](#) (const [CompositeMarkupNode](#) &node)=0

### Amis

- class [TextNode](#)
- class [MarkupNode](#)
- class [CompositeMarkupNode](#)

#### 6.3.1 Description détaillée

Définition à la ligne 19 du fichier InterfaceNodeVisitor.hpp.

#### 6.3.2 Documentation des constructeurs et destructeur

6.3.2.1 virtual xml : :InterfaceNodeVisitor : :~InterfaceNodeVisitor ( ) [inline, virtual]

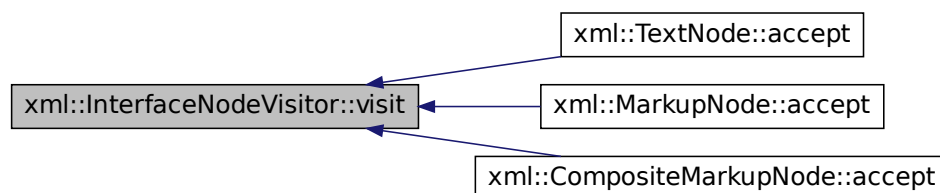
Définition à la ligne 22 du fichier InterfaceNodeVisitor.hpp.

#### 6.3.3 Documentation des fonctions membres

6.3.3.1 virtual void xml : :InterfaceNodeVisitor : :visit ( const TextNode & node )  
[protected, pure virtual]

Implémenté dans [xml : :DotOutputVisitor](#), et [xml : :OutputVisitor](#).

Voici le graphe d'appel pour cette fonction :



**6.3.3.2** `virtual void xml : :InterfaceNodeVisitor : :visit ( const CompositeMarkupNode & node )`  
[protected, pure virtual]

Implémenté dans [xml : :DotOutputVisitor](#), et [xml : :OutputVisitor](#).

**6.3.3.3** `virtual void xml : :InterfaceNodeVisitor : :visit ( const MarkupNode & node )`  
[protected, pure virtual]

Implémenté dans [xml : :DotOutputVisitor](#), et [xml : :OutputVisitor](#).

## 6.3.4 Documentation des fonctions amies et associées

**6.3.4.1** `friend class CompositeMarkupNode` [friend]

Définition à la ligne 34 du fichier InterfaceNodeVisitor.hpp.

**6.3.4.2** `friend class MarkupNode` [friend]

Définition à la ligne 33 du fichier InterfaceNodeVisitor.hpp.

**6.3.4.3** `friend class TextNode` [friend]

Définition à la ligne 32 du fichier InterfaceNodeVisitor.hpp.

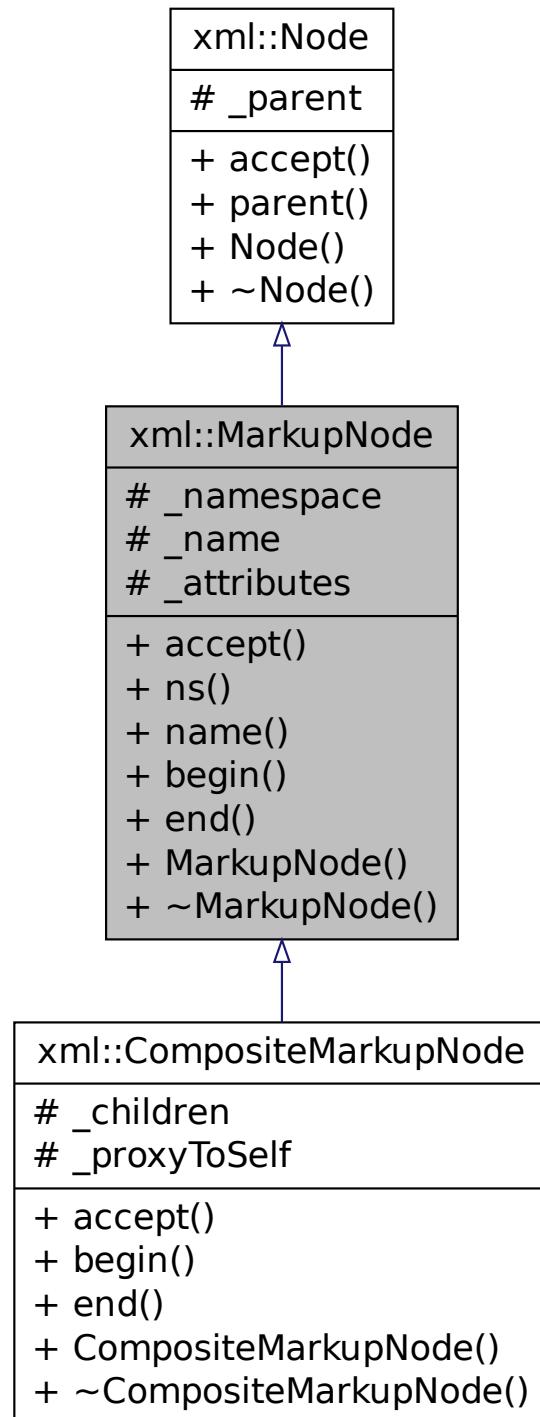
La documentation de cette classe a été générée à partir du fichier suivant :

– src/[InterfaceNodeVisitor.hpp](#)

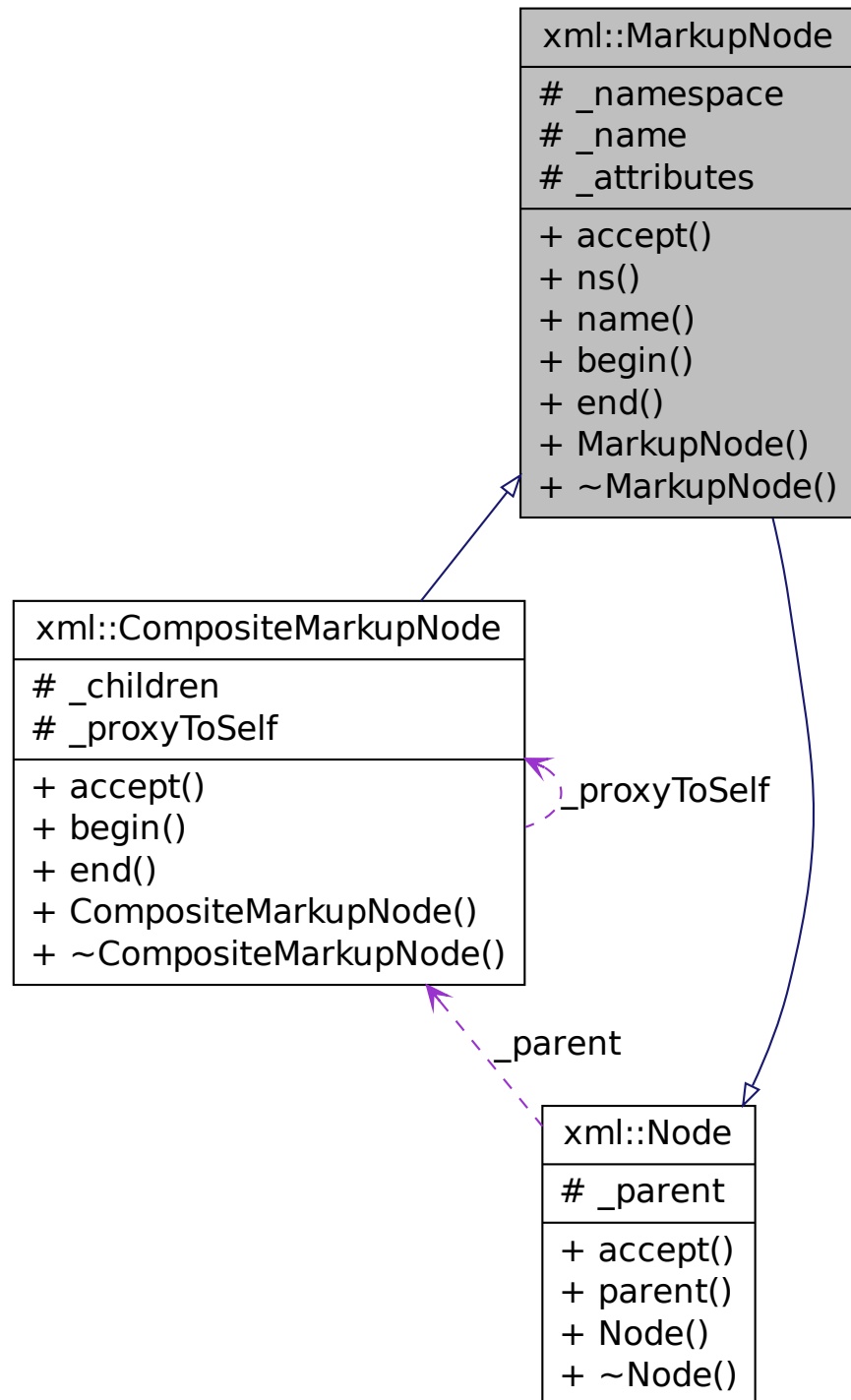
## 6.4 Référence de la classe xml : :MarkupNode

```
#include <MarkupNode.hh>
```

Graphe d'héritage de xml : :MarkupNode :



Graphe de collaboration de xml : :MarkupNode :





### Types publics

- typedef std : :map< std : :string, std : :string > [Attributes](#)
- typedef \_Attributes : :const\_iterator [AttributesIterator](#)

### Fonctions membres publiques

- virtual void [accept](#) ([InterfaceNodeVisitor](#) &visitor) const
- std : :string [ns](#) () const
- std : :string [name](#) () const
- [AttributesIterator](#) [begin](#) () const
- [AttributesIterator](#) [end](#) () const
- [MarkupNode](#) ([CompositeMarkupNode](#) \*\*parent, const std : :string &ns, const std : :string &name, const [Attributes](#) &attributes)
- virtual [~MarkupNode](#) ()

### Types protégés

- typedef std : :map< std : :string, std : :string > [\\_Attributes](#)

### Attributs protégés

- std : :string [\\_namespace](#)
- std : :string [\\_name](#)
- [\\_Attributes](#) [\\_attributes](#)

#### 6.4.1 Description détaillée

Définition à la ligne 26 du fichier MarkupNode.hh.

#### 6.4.2 Documentation des définitions de type membres

**6.4.2.1** typedef std : :map<std : :string, std : :string> xml : :MarkupNode : :\_Attributes  
[protected]

Définition à la ligne 29 du fichier MarkupNode.hh.

**6.4.2.2** typedef std : :map<std : :string, std : :string> xml : :MarkupNode : :Attributes

Définition à la ligne 35 du fichier MarkupNode.hh.

**6.4.2.3** typedef \_Attributes : :const\_iterator xml : :MarkupNode : :AttributesIterator

Définition à la ligne 36 du fichier MarkupNode.hh.

### 6.4.3 Documentation des constructeurs et destructeur

6.4.3.1 `xml : :MarkupNode : :MarkupNode ( CompositeMarkupNode ** parent, const std : :string & ns, const std : :string & name, const Attributes & attributes )`

Définition à la ligne 61 du fichier MarkupNode.cpp.

6.4.3.2 `xml : :MarkupNode : :~MarkupNode ( ) [virtual]`

Définition à la ligne 69 du fichier MarkupNode.cpp.

### 6.4.4 Documentation des fonctions membres

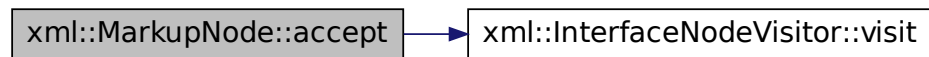
6.4.4.1 `void xml : :MarkupNode : :accept ( InterfaceNodeVisitor & visitor ) const [virtual]`

Implémente [xml : :Node](#).

Réimplémentée dans [xml : :CompositeMarkupNode](#).

Définition à la ligne 32 du fichier MarkupNode.cpp.

Voici le graphe d'appel pour cette fonction :



6.4.4.2 `MarkupNode : :AttributesIterator xml : :MarkupNode : :begin ( ) const`

Réimplémentée dans [xml : :CompositeMarkupNode](#).

Définition à la ligne 47 du fichier MarkupNode.cpp.

6.4.4.3 `MarkupNode : :AttributesIterator xml : :MarkupNode : :end ( ) const`

Réimplémentée dans [xml : :CompositeMarkupNode](#).

Définition à la ligne 52 du fichier MarkupNode.cpp.

6.4.4.4 `string xml : :MarkupNode : :name ( ) const`

Définition à la ligne 42 du fichier MarkupNode.cpp.

#### 6.4.4.5 string xml : :MarkupNode : :ns ( ) const

Définition à la ligne 37 du fichier MarkupNode.cpp.

### 6.4.5 Documentation des données membres

#### 6.4.5.1 \_Attributes xml : :MarkupNode : :\_attributes [protected]

Définition à la ligne 102 du fichier MarkupNode.hh.

#### 6.4.5.2 std : :string xml : :MarkupNode : :\_name [protected]

Définition à la ligne 101 du fichier MarkupNode.hh.

#### 6.4.5.3 std : :string xml : :MarkupNode : :\_namespace [protected]

Définition à la ligne 100 du fichier MarkupNode.hh.

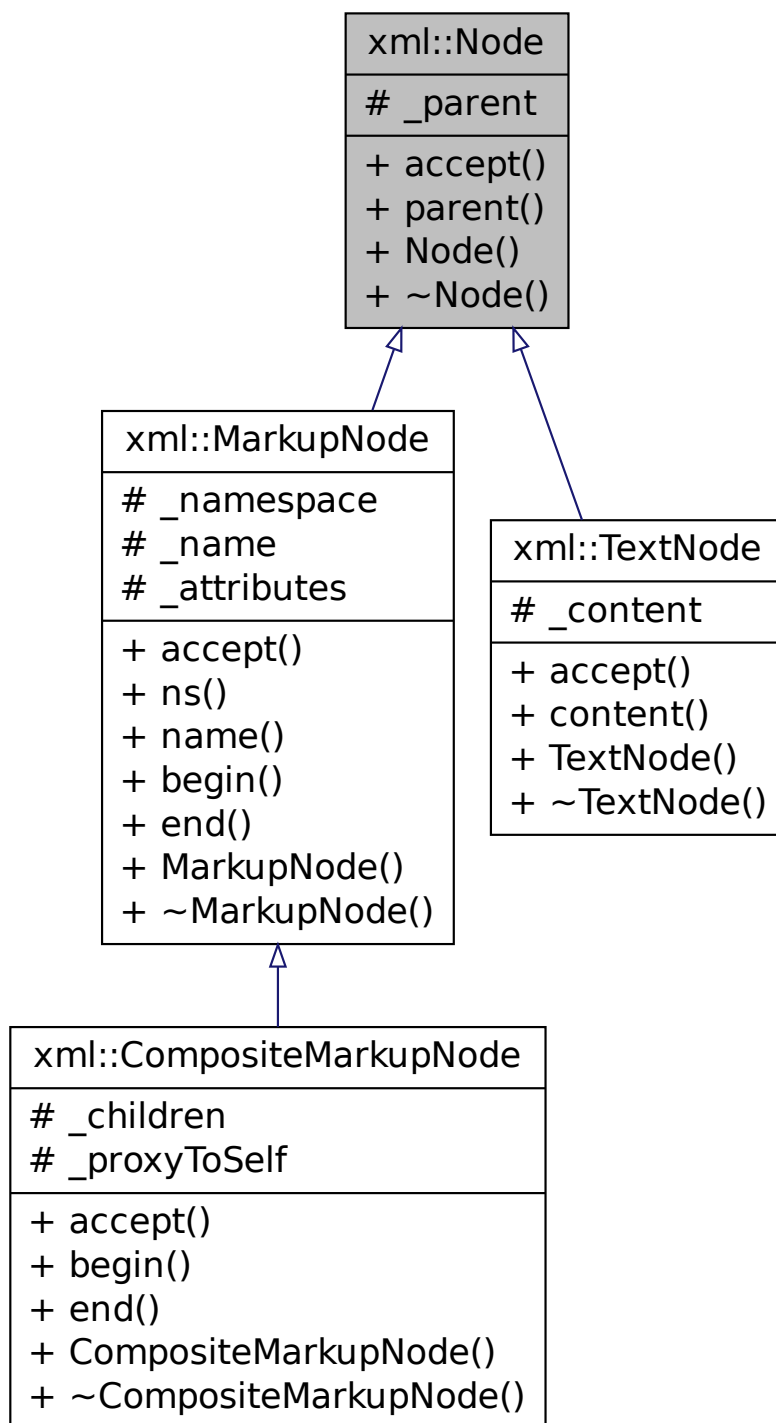
La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[MarkupNode.hh](#)
- src/[MarkupNode.cpp](#)

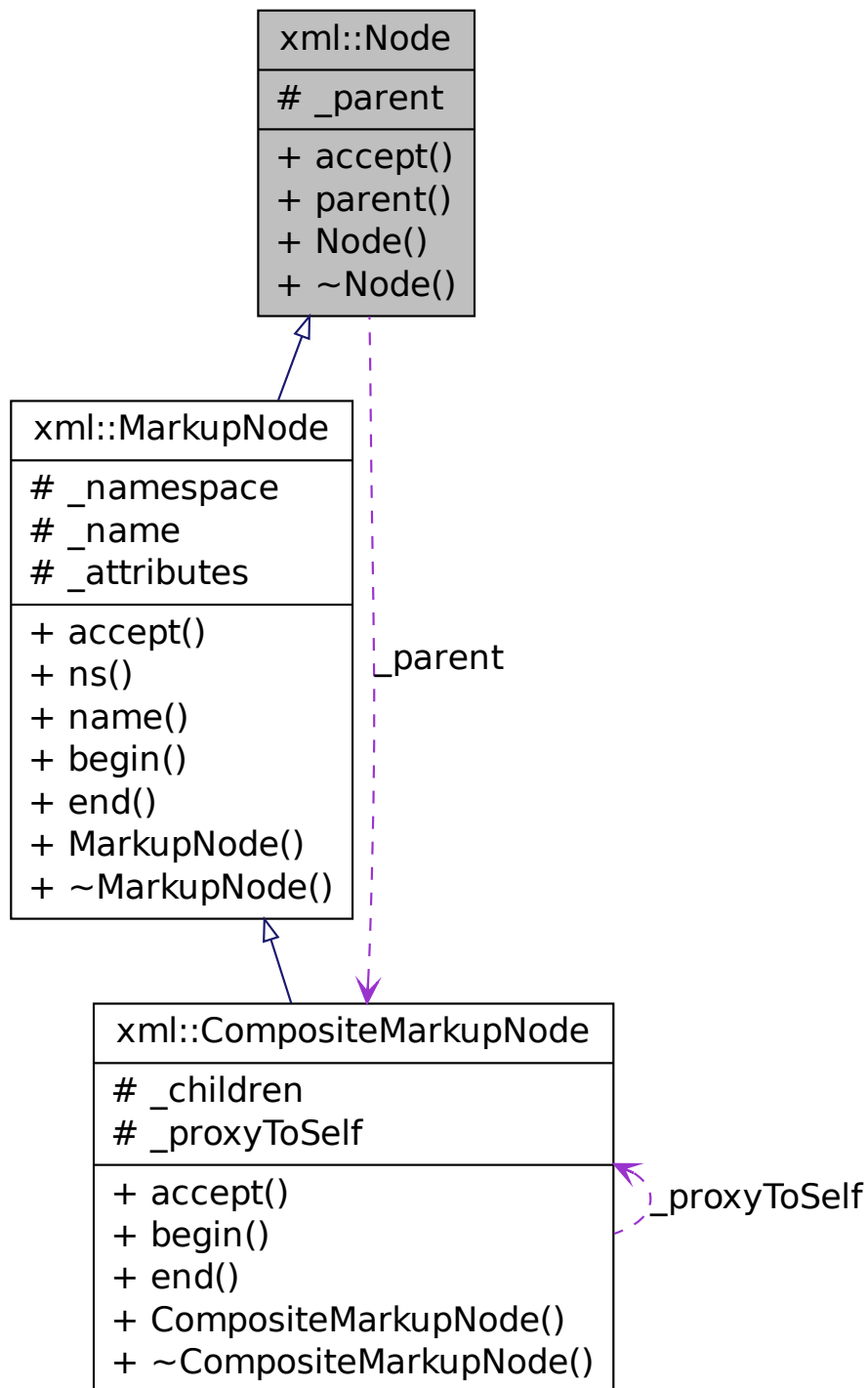
## 6.5 Référence de la classe xml : :Node

```
#include <Node.hh>
```

Graphes d'héritage de xml : :Node :



Graphe de collaboration de xml : :Node :



### Fonctions membres publiques

- virtual void `accept` (`InterfaceNodeVisitor` &visitor) const =0
- `Node` \* `parent` () const
- `Node` (`CompositeMarkupNode` \*\*parent)
- virtual `~Node` ()

### Attributs protégés

- `CompositeMarkupNode` \*\* `_parent`

#### 6.5.1 Description détaillée

Définition à la ligne 26 du fichier Node.hh.

#### 6.5.2 Documentation des constructeurs et destructeur

##### 6.5.2.1 `xml : :Node : :Node ( CompositeMarkupNode ** parent )`

Définition à la ligne 45 du fichier Node.cpp.

##### 6.5.2.2 `xml : :Node : :~Node ( ) [virtual]`

Définition à la ligne 51 du fichier Node.cpp.

#### 6.5.3 Documentation des fonctions membres

##### 6.5.3.1 `virtual void xml : :Node : :accept ( InterfaceNodeVisitor & visitor ) const [pure virtual]`

Implémenté dans `xml : :CompositeMarkupNode`, `xml : :MarkupNode`, et `xml : :TextNode`.

##### 6.5.3.2 `Node * xml : :Node : :parent ( ) const`

Définition à la ligne 33 du fichier Node.cpp.

#### 6.5.4 Documentation des données membres

##### 6.5.4.1 `CompositeMarkupNode** xml : :Node : :_parent [protected]`

Définition à la ligne 67 du fichier Node.hh.

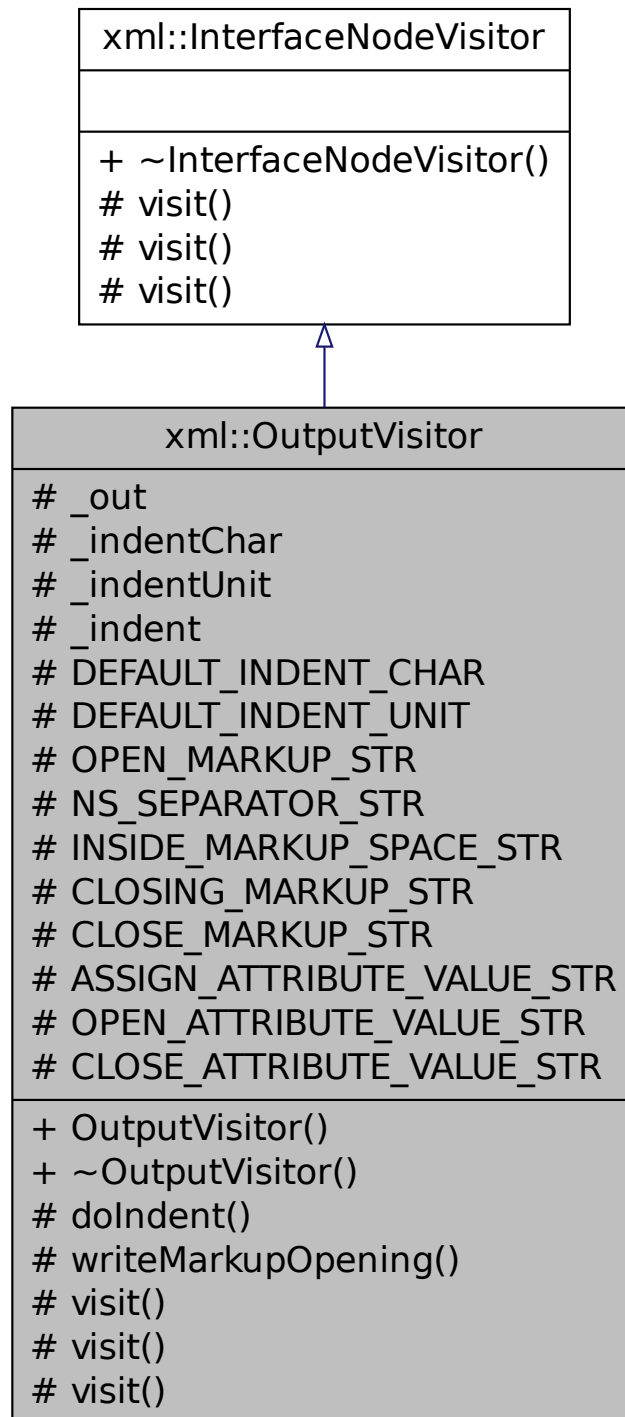
La documentation de cette classe a été générée à partir des fichiers suivants :

- `src/Node.hh`
- `src/Node.cpp`

## 6.6 Référence de la classe xml : :OutputVisitor

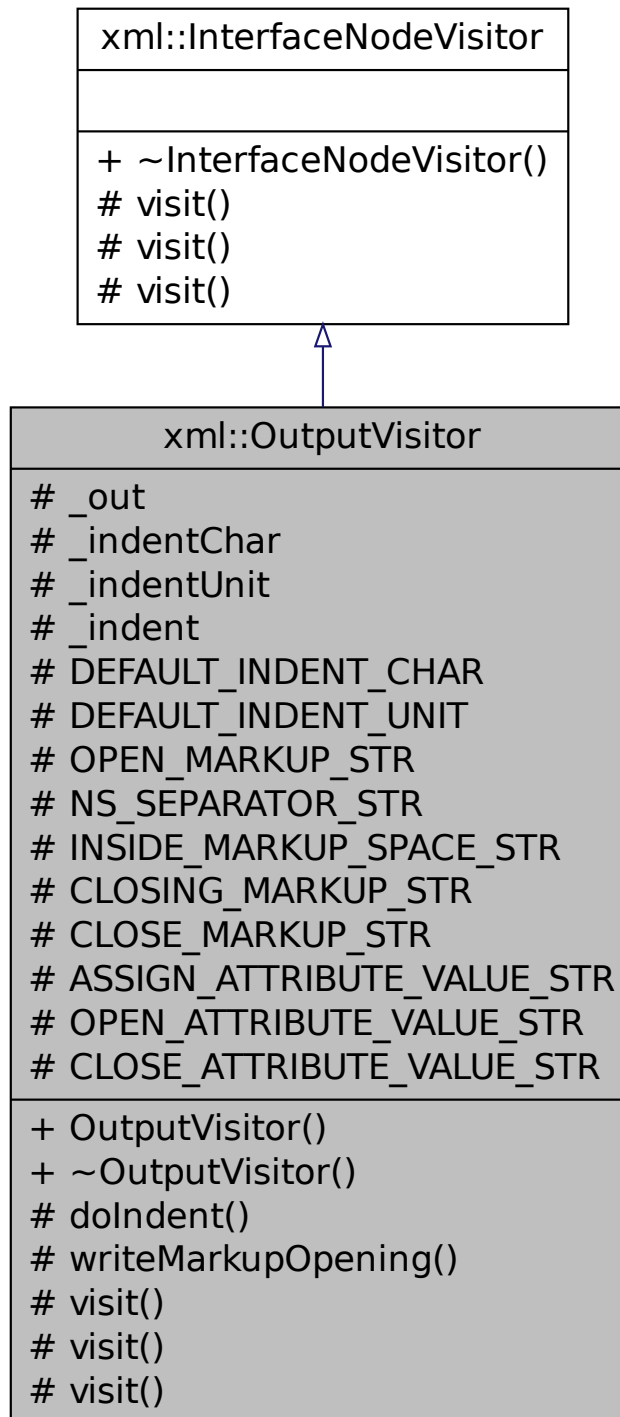
```
#include <OutputVisitor.hh>
```

Graphe d'héritage de xml : :OutputVisitor :





Graphe de collaboration de xml : :OutputVisitor :



## Fonctions membres publiques

- `OutputVisitor` (`std::ostream &out`, `char indentChar=DEFAULT_INDENT_CHAR`, `unsigned int indentUnit=DEFAULT_INDENT_UNIT`)
- virtual `~OutputVisitor` ()

## Fonctions membres protégées

- void `doIndent` ()
- void `writeMarkupOpening` (const `MarkupNode` &node)
- virtual void `visit` (const `TextNode` &node)
- virtual void `visit` (const `MarkupNode` &node)
- virtual void `visit` (const `CompositeMarkupNode` &node)

## Attributs protégés

- `std::ostream &_out`
- `char _indentChar`
- `unsigned _indentUnit`
- `unsigned int _indent`

## Attributs protégés statiques

- static const `char DEFAULT_INDENT_CHAR = ' '`
- static const unsigned `char DEFAULT_INDENT_UNIT = 1`
- static const `std::string OPEN_MARKUP_STR = "<"`
- static const `std::string NS_SEPARATOR_STR = ":"`
- static const `std::string INSIDE_MARKUP_SPACE_STR = " "`
- static const `std::string CLOSING_MARKUP_STR = "/"`
- static const `std::string CLOSE_MARKUP_STR = ">"`
- static const `std::string ASSIGN_ATTRIBUTE_VALUE_STR = "="`
- static const `std::string OPEN_ATTRIBUTE_VALUE_STR = "\""`
- static const `std::string CLOSE_ATTRIBUTE_VALUE_STR = "\""`

### 6.6.1 Description détaillée

Définition à la ligne 24 du fichier `OutputVisitor.hh`.

### 6.6.2 Documentation des constructeurs et destructeur

**6.6.2.1** `xml::OutputVisitor::OutputVisitor ( std::ostream & out, char indentChar = DEFAULT_INDENT_CHAR, unsigned int indentUnit = DEFAULT_INDENT_UNIT )`

**6.6.2.2** `xml::OutputVisitor::~OutputVisitor ( ) [virtual]`

Définition à la ligne 53 du fichier `OutputVisitor.cpp`.

### 6.6.3 Documentation des fonctions membres

#### 6.6.3.1 `void xml : :OutputVisitor : :doIndent ( )` [protected]

Définition à la ligne 63 du fichier `OutputVisitor.cpp`.

#### 6.6.3.2 `void xml : :OutputVisitor : :visit ( const CompositeMarkupNode & node )` [protected, virtual]

Implémente [xml : :InterfaceNodeVisitor](#).

Définition à la ligne 104 du fichier `OutputVisitor.cpp`.

#### 6.6.3.3 `void xml : :OutputVisitor : :visit ( const MarkupNode & node )` [protected, virtual]

Implémente [xml : :InterfaceNodeVisitor](#).

Définition à la ligne 96 du fichier `OutputVisitor.cpp`.

Voici le graphe d'appel pour cette fonction :



#### 6.6.3.4 `void xml : :OutputVisitor : :visit ( const TextNode & node )` [protected, virtual]

Implémente [xml : :InterfaceNodeVisitor](#).

Définition à la ligne 91 du fichier `OutputVisitor.cpp`.

#### 6.6.3.5 `void xml : :OutputVisitor : :writeMarkupOpening ( const MarkupNode & node )` [protected]

Définition à la ligne 73 du fichier `OutputVisitor.cpp`.

## 6.6.4 Documentation des données membres

**6.6.4.1** `unsigned int xml : :OutputVisitor : :_indent` `[protected]`

Définition à la ligne 68 du fichier OutputVisitor.hh.

**6.6.4.2** `char xml : :OutputVisitor : :_indentChar` `[protected]`

Définition à la ligne 66 du fichier OutputVisitor.hh.

**6.6.4.3** `unsigned xml : :OutputVisitor : :_indentUnit` `[protected]`

Définition à la ligne 67 du fichier OutputVisitor.hh.

**6.6.4.4** `std : :ostream& xml : :OutputVisitor : :_out` `[protected]`

Définition à la ligne 65 du fichier OutputVisitor.hh.

**6.6.4.5** `const std : :string xml : :OutputVisitor : :ASSIGN_ATTRIBUTE_VALUE_STR = "="`  
`[static, protected]`

Définition à la ligne 61 du fichier OutputVisitor.hh.

**6.6.4.6** `const std : :string xml : :OutputVisitor : :CLOSE_ATTRIBUTE_VALUE_STR = "\""`  
`[static, protected]`

Définition à la ligne 63 du fichier OutputVisitor.hh.

**6.6.4.7** `const std : :string xml : :OutputVisitor : :CLOSE_MARKUP_STR = ">"` `[static, protected]`

Définition à la ligne 60 du fichier OutputVisitor.hh.

**6.6.4.8** `const std : :string xml : :OutputVisitor : :CLOSING_MARKUP_STR = "/"` `[static, protected]`

Définition à la ligne 59 du fichier OutputVisitor.hh.

**6.6.4.9** `const char xml : :OutputVisitor : :DEFAULT_INDENT_CHAR = ' '` `[static, protected]`

Définition à la ligne 27 du fichier OutputVisitor.hh.

**6.6.4.10** `const unsigned char xml : :OutputVisitor : :DEFAULT_INDENT_UNIT = 1` `[static, protected]`

Définition à la ligne 28 du fichier OutputVisitor.hh.

**6.6.4.11** `const std : :string xml : :OutputVisitor : :INSIDE_MARKUP_SPACE_STR = " "` `[static, protected]`

Définition à la ligne 58 du fichier OutputVisitor.hh.

**6.6.4.12** `const std : :string xml : :OutputVisitor : :NS_SEPARATOR_STR = ":"` `[static, protected]`

Définition à la ligne 57 du fichier OutputVisitor.hh.

**6.6.4.13** `const std : :string xml : :OutputVisitor : :OPEN_ATTRIBUTE_VALUE_STR = "\""` `[static, protected]`

Définition à la ligne 62 du fichier OutputVisitor.hh.

**6.6.4.14** `const std : :string xml : :OutputVisitor : :OPEN_MARKUP_STR = "<"` `[static, protected]`

Définition à la ligne 56 du fichier OutputVisitor.hh.

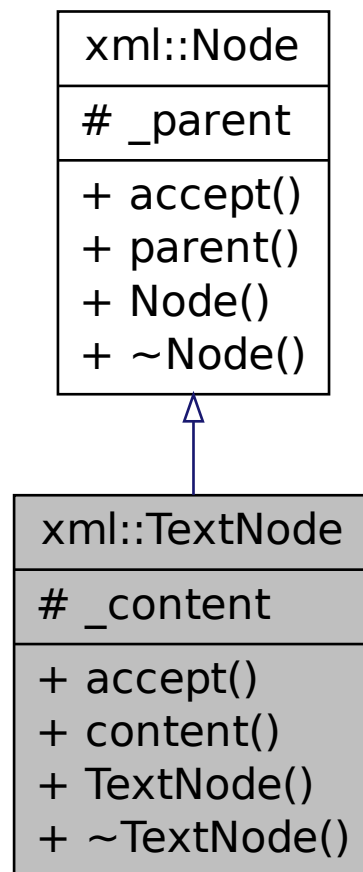
La documentation de cette classe a été générée à partir des fichiers suivants :

- [src/OutputVisitor.hh](#)
- [src/OutputVisitor.cpp](#)

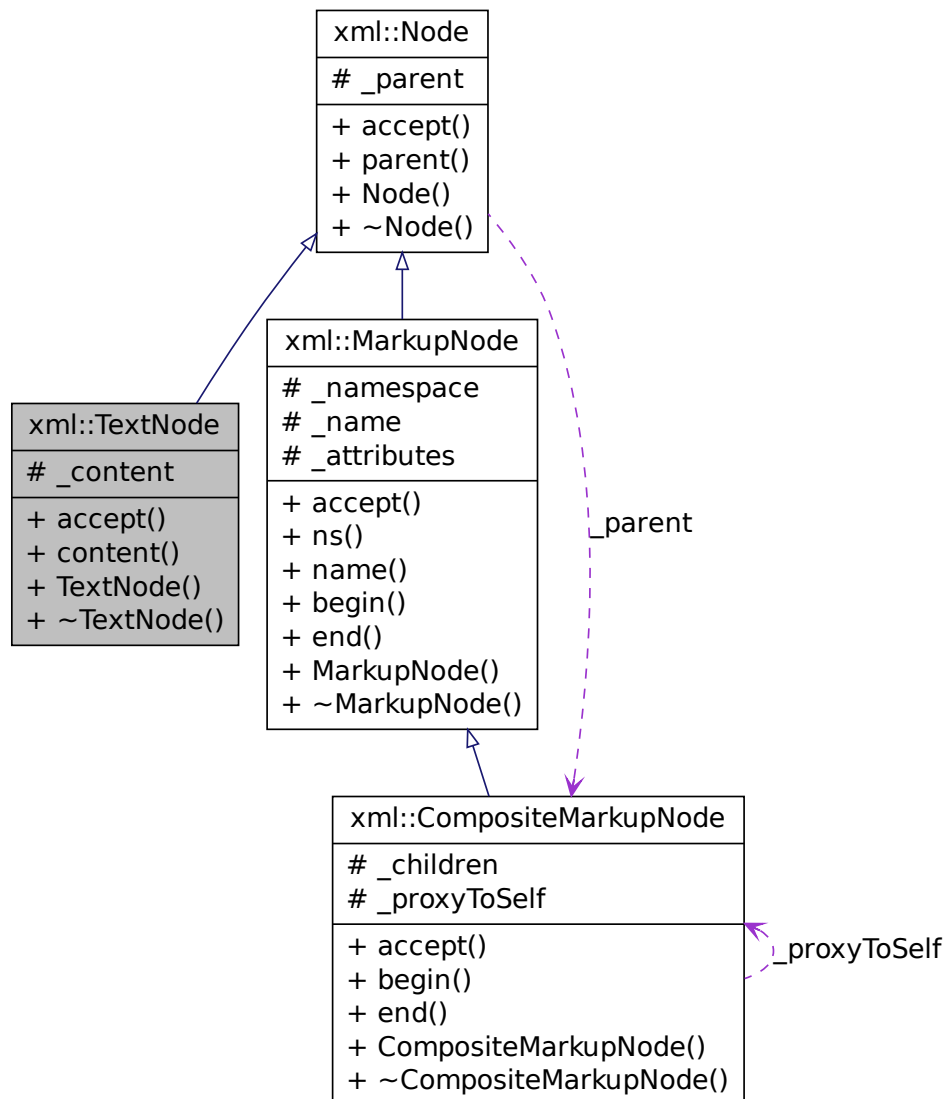
## 6.7 Référence de la classe xml : :TextNode

```
#include <TextNode.hh>
```

Graphe d'héritage de xml::TextNode :



Graphe de collaboration de xml : :TextNode :



### Fonctions membres publiques

- virtual void `accept` (`InterfaceNodeVisitor` &visitor) const
- std : :string `content` () const
- `TextNode` (`CompositeMarkupNode` \*\*parent, const std : :string &content)
- virtual `~TextNode` ()

## Attributs protégés

– std : :string [\\_content](#)

### 6.7.1 Description détaillée

Définition à la ligne 25 du fichier XmlNode.hh.

### 6.7.2 Documentation des constructeurs et destructeur

6.7.2.1 `xml : :XmlNode : :XmlNode ( CompositeMarkupNode ** parent, const std : :string & content )`

6.7.2.2 `xml : :XmlNode : :~XmlNode ( ) [virtual]`

Définition à la ligne 53 du fichier XmlNode.cpp.

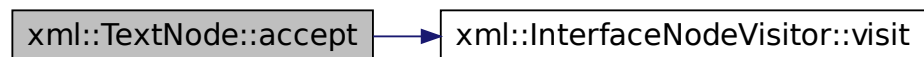
### 6.7.3 Documentation des fonctions membres

6.7.3.1 `void xml : :XmlNode : :accept ( InterfaceNodeVisitor & visitor ) const [virtual]`

Implémente [xml : :Node](#).

Définition à la ligne 32 du fichier XmlNode.cpp.

Voici le graphe d'appel pour cette fonction :



6.7.3.2 `string xml : :XmlNode : :content ( ) const`

Définition à la ligne 37 du fichier XmlNode.cpp.



Voici le graphe d'appel pour cette fonction :



## 6.7.4 Documentation des données membres

### 6.7.4.1 `std : :string xml : :TextNode : :_content` [protected]

Définition à la ligne 62 du fichier TextNode.hh.

La documentation de cette classe a été générée à partir des fichiers suivants :

- src/[TextNode.hh](#)
- src/[TextNode.cpp](#)



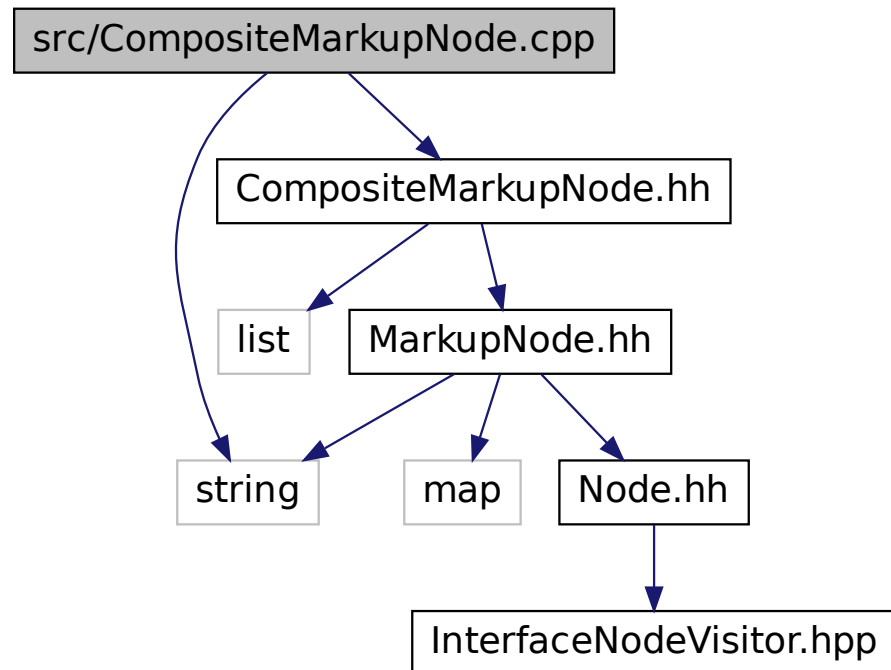
## Chapitre 7

# Documentation des fichiers

### 7.1 Référence du fichier src/CompositeMarkupNode.cpp

```
#include <string>
#include "CompositeMarkupNode.hh"
```

Graphe des dépendances par inclusion de CompositeMarkupNode.cpp :



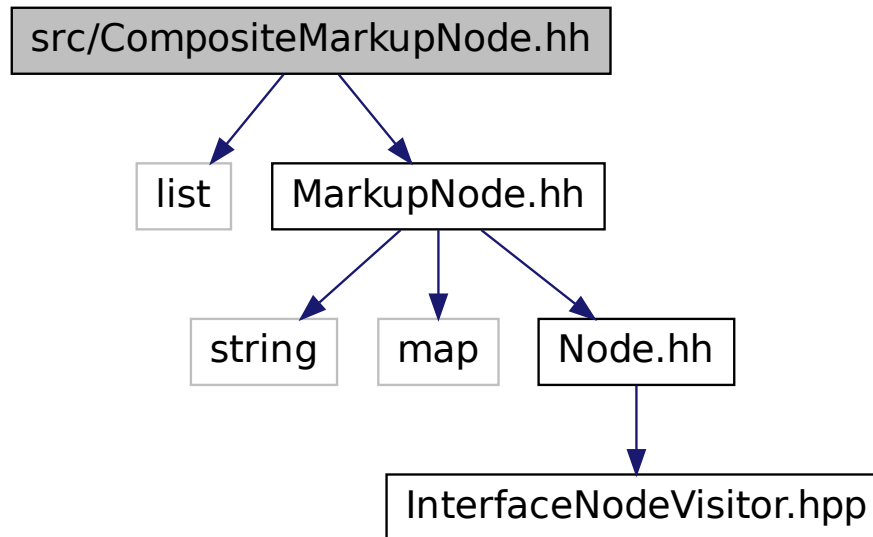
### Espaces de nommage

– namespace `xml`

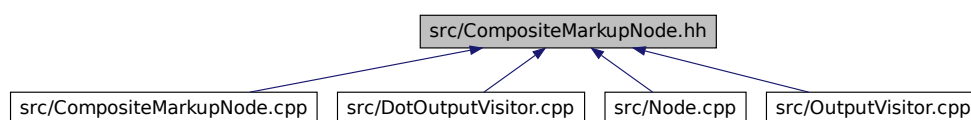
## 7.2 Référence du fichier `src/CompositeMarkupNode.hh`

```
#include <list>
#include "MarkupNode.hh"
```

Graphe des dépendances par inclusion de CompositeMarkupNode.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `xml` : `:CompositeMarkupNode`

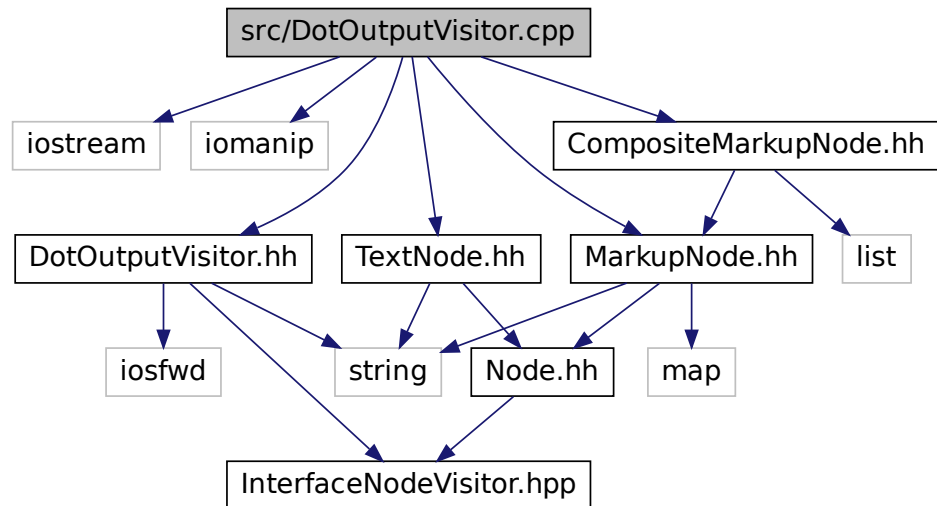
## Espaces de nommage

– namespace `xml`

### 7.3 Référence du fichier src/DotOutputVisitor.cpp

```
#include <iostream>
#include <iomanip>
#include "DotOutputVisitor.hh"
#include "TextNode.hh"
#include "MarkupNode.hh"
#include "CompositeMarkupNode.hh"
```

Graphes des dépendances par inclusion de DotOutputVisitor.cpp :



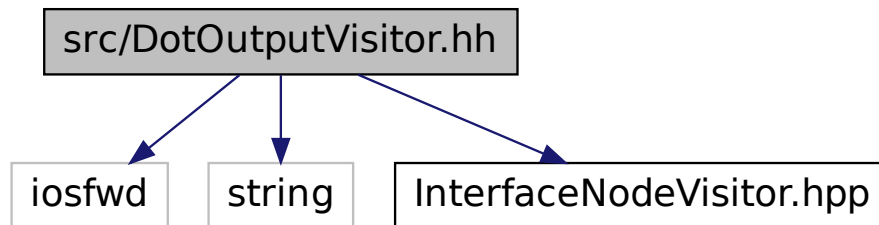
#### Espaces de nommage

– namespace `xml`

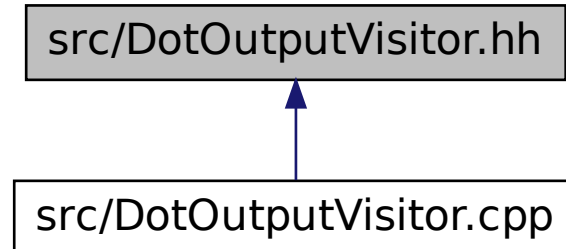
### 7.4 Référence du fichier src/DotOutputVisitor.hh

```
#include <iosfwd>
#include <string>
#include "InterfaceNodeVisitor.hpp"
```

Graphe des dépendances par inclusion de DotOutputVisitor.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

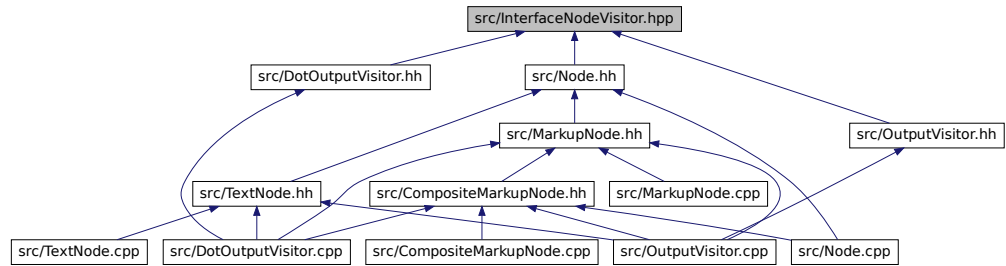
– class `xml : :DotOutputVisitor`

### Espaces de nommage

– namespace `xml`

## 7.5 Référence du fichier src/InterfaceNodeVisitor.hpp

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `xml : InterfaceNodeVisitor`

### Espaces de nommage

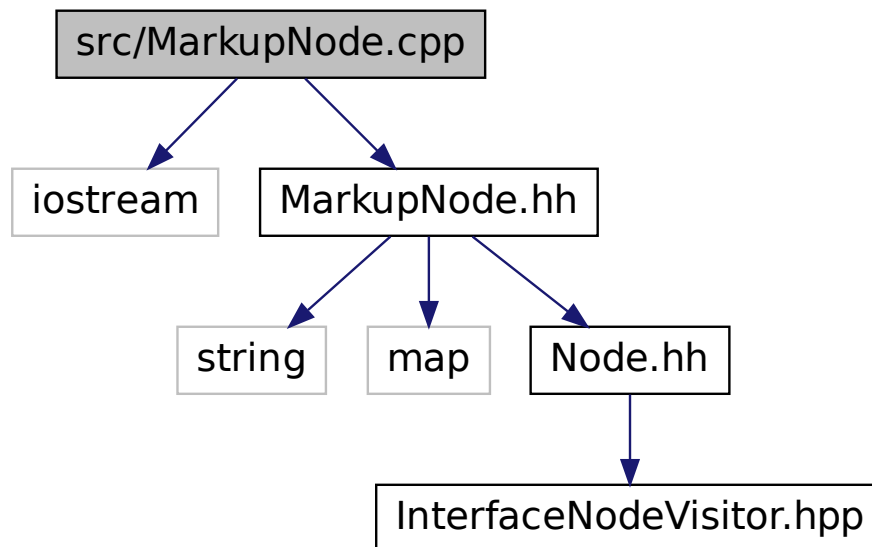
– namespace `xml`

## 7.6 Référence du fichier src/MarkupNode.cpp

```
#include <iostream>
#include "MarkupNode.hh"
```



Graphe des dépendances par inclusion de MarkupNode.cpp :



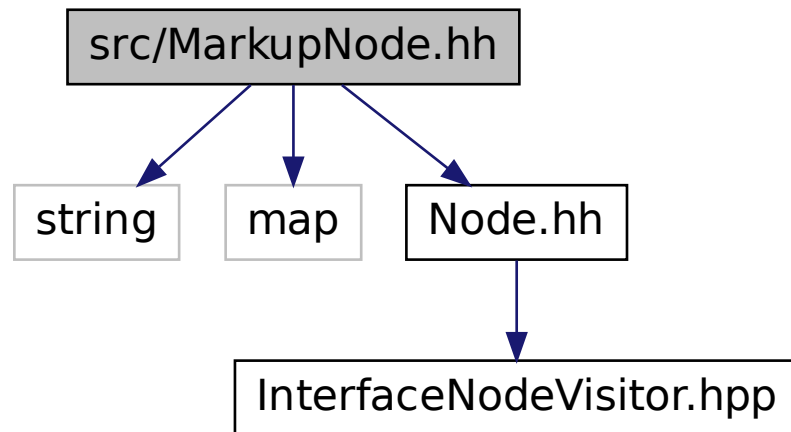
### Espaces de nommage

– namespace `xml`

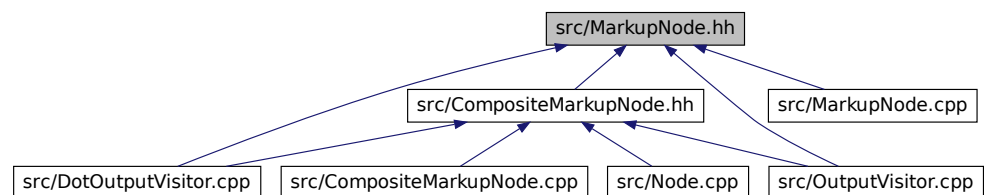
## 7.7 Référence du fichier src/MarkupNode.hh

```
#include <string>
#include <map>
#include "Node.hh"
```

Graphe des dépendances par inclusion de MarkupNode.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `xml` : `MarkupNode`

## Espaces de nommage

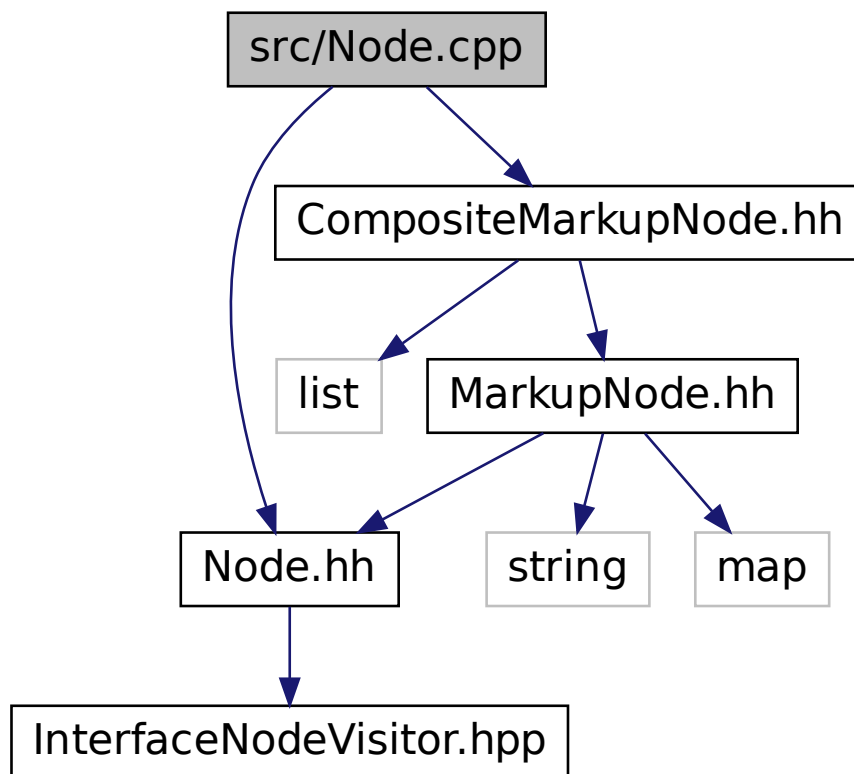
– namespace `xml`

## 7.8 Référence du fichier src/Node.cpp

```
#include "Node.hh"
```

```
#include "CompositeMarkupNode.hh"
```

Graphe des dépendances par inclusion de Node.cpp :



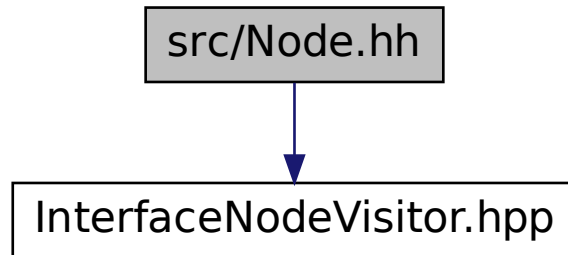
### Espaces de nommage

– namespace `xml`

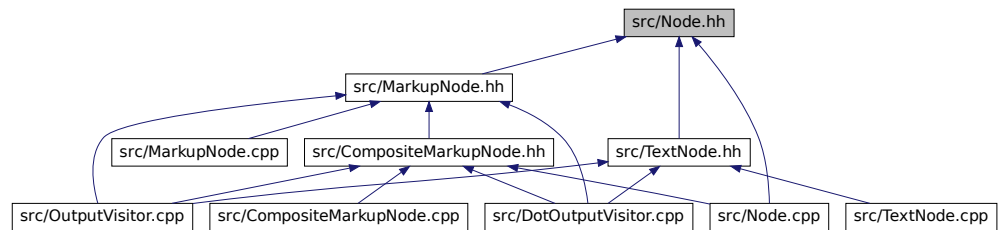
## 7.9 Référence du fichier src/Node.hh

```
#include "InterfaceNodeVisitor.hpp"
```

Graphe des dépendances par inclusion de Node.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `xml : :Node`

## Espaces de nommage

– namespace `xml`

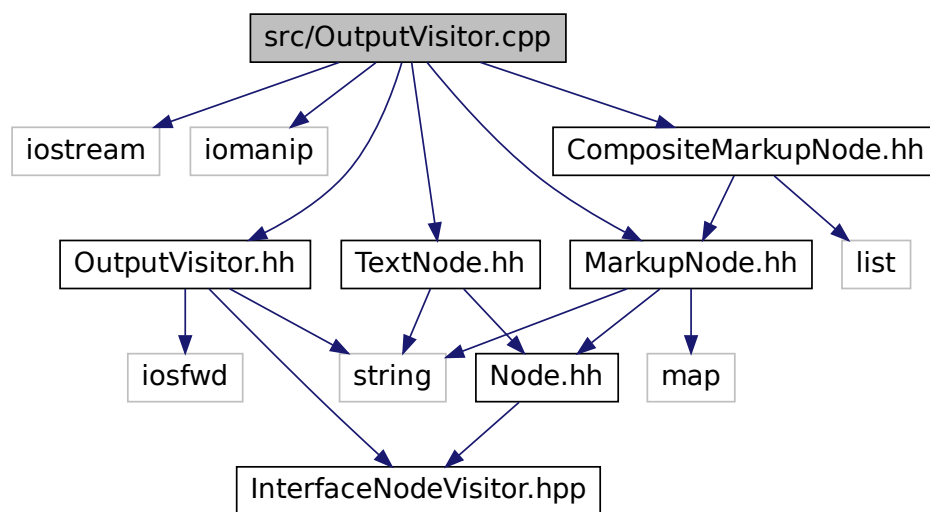
## 7.10 Référence du fichier src/OutputVisitor.cpp

```

#include <iostream>
#include <iomanip>
#include "OutputVisitor.hh"
  
```

```
#include "TextNode.hh"
#include "MarkupNode.hh"
#include "CompositeMarkupNode.hh"
```

Graphe des dépendances par inclusion de OutputVisitor.cpp :



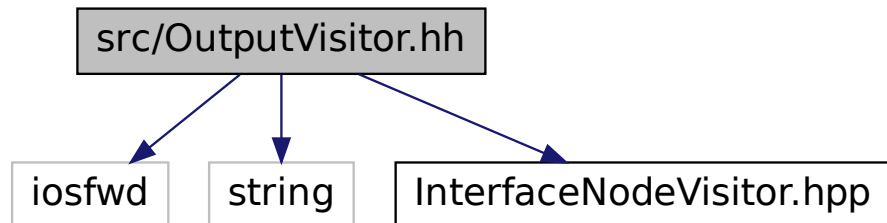
## Espaces de nommage

– namespace `xml`

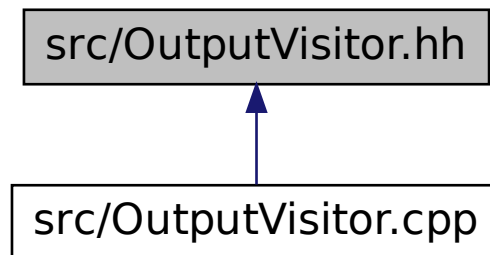
## 7.11 Référence du fichier src/OutputVisitor.hh

```
#include <iosfwd>
#include <string>
#include "InterfaceNodeVisitor.hpp"
```

Graphe des dépendances par inclusion de OutputVisitor.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

– class `xml : :OutputVisitor`

### Espaces de nommage

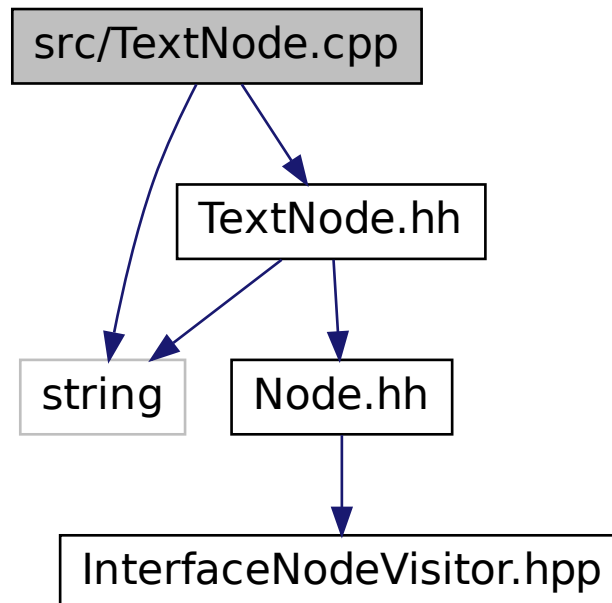
– namespace `xml`

## 7.12 Référence du fichier src/TextNode.cpp

```
#include <string>
```

```
#include "TextNode.hh"
```

Graphe des dépendances par inclusion de TextNode.cpp :



### Espaces de nommage

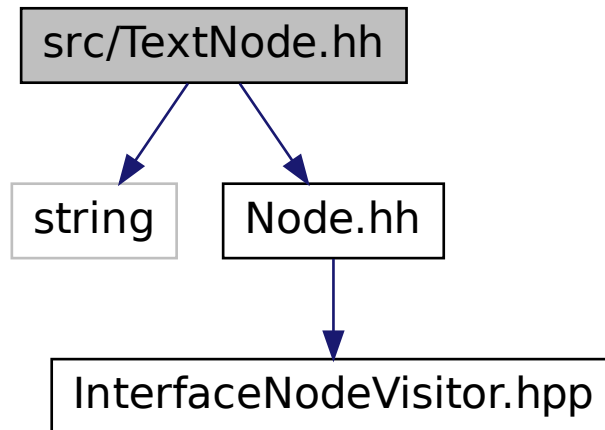
– namespace `xml`

## 7.13 Référence du fichier src/TextNode.hh

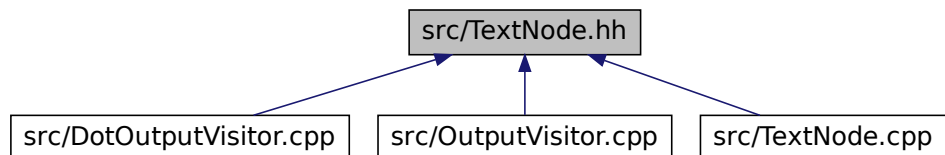
```
#include <string>
```

```
#include "Node.hh"
```

Graphe des dépendances par inclusion de TextNode.hh :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

– class `xml : :TextNode`

## Espaces de nommage

– namespace `xml`



# Index

- ~CompositeMarkupNode
  - xml : :CompositeMarkupNode, [15](#)
- ~DotOutputVisitor
  - xml : :DotOutputVisitor, [19](#)
- ~InterfaceNodeVisitor
  - xml : :InterfaceNodeVisitor, [23](#)
- ~MarkupNode
  - xml : :MarkupNode, [28](#)
- ~Node
  - xml : :Node, [32](#)
- ~OutputVisitor
  - xml : :OutputVisitor, [36](#)
- ~TextNode
  - xml : :TextNode, [42](#)
- \_Attributes
  - xml : :MarkupNode, [27](#)
- \_Children
  - xml : :CompositeMarkupNode, [14](#)
- \_attributes
  - xml : :MarkupNode, [29](#)
- \_children
  - xml : :CompositeMarkupNode, [16](#)
- \_content
  - xml : :TextNode, [43](#)
- \_document
  - xml : :DotOutputVisitor, [20](#)
- \_indent
  - xml : :OutputVisitor, [38](#)
- \_indentChar
  - xml : :OutputVisitor, [38](#)
- \_indentUnit
  - xml : :OutputVisitor, [38](#)
- \_name
  - xml : :MarkupNode, [29](#)
- \_namespace
  - xml : :MarkupNode, [29](#)
- \_nodesCounter
  - xml : :DotOutputVisitor, [20](#)
- \_out
  - xml : :DotOutputVisitor, [21](#)
  - xml : :OutputVisitor, [38](#)
- \_parent
  - xml : :Node, [32](#)
- \_proxyToSelf
  - xml : :CompositeMarkupNode, [16](#)
- accept
  - xml : :CompositeMarkupNode, [15](#)
  - xml : :MarkupNode, [28](#)
  - xml : :Node, [32](#)
  - xml : :TextNode, [42](#)
- ASSIGN\_ATTRIBUTE\_VALUE\_STR
  - xml : :OutputVisitor, [38](#)
- Attributes
  - xml : :MarkupNode, [27](#)
- AttributesIterator
  - xml : :MarkupNode, [27](#)
- begin
  - xml : :CompositeMarkupNode, [15](#)
  - xml : :MarkupNode, [28](#)
- Children
  - xml : :CompositeMarkupNode, [14](#)
- ChildrenIterator
  - xml : :CompositeMarkupNode, [14](#)
- CLOSE\_ATTRIBUTE\_VALUE\_STR
  - xml : :OutputVisitor, [38](#)
- CLOSE\_MARKUP\_STR
  - xml : :OutputVisitor, [38](#)
- CLOSING\_MARKUP\_STR
  - xml : :OutputVisitor, [38](#)
- CompositeMarkupNode
  - xml : :CompositeMarkupNode, [15](#)
  - xml : :InterfaceNodeVisitor, [24](#)
- content
  - xml : :TextNode, [42](#)
- declareNode
  - xml : :DotOutputVisitor, [20](#)
- DEFAULT\_INDENT\_CHAR
  - xml : :OutputVisitor, [38](#)
- DEFAULT\_INDENT\_UNIT

- xml : :OutputVisitor, [38](#)
- doIndent
  - xml : :OutputVisitor, [37](#)
- DotOutputVisitor
  - xml : :DotOutputVisitor, [19](#)
- end
  - xml : :CompositeMarkupNode, [15](#)
  - xml : :MarkupNode, [28](#)
- GRAPH\_TYPE
  - xml : :DotOutputVisitor, [21](#)
- INSIDE\_MARKUP\_SPACE\_STR
  - xml : :OutputVisitor, [39](#)
- LINK
  - xml : :DotOutputVisitor, [21](#)
- MarkupNode
  - xml : :InterfaceNodeVisitor, [24](#)
  - xml : :MarkupNode, [28](#)
- name
  - xml : :MarkupNode, [28](#)
- Node
  - xml : :Node, [32](#)
- ns
  - xml : :MarkupNode, [28](#)
- NS\_SEPARATOR\_STR
  - xml : :OutputVisitor, [39](#)
- OPEN\_ATTRIBUTE\_VALUE\_STR
  - xml : :OutputVisitor, [39](#)
- OPEN\_MARKUP\_STR
  - xml : :OutputVisitor, [39](#)
- OutputVisitor
  - xml : :OutputVisitor, [36](#)
- parent
  - xml : :Node, [32](#)
- src/CompositeMarkupNode.cpp, [45](#)
- src/CompositeMarkupNode.hh, [46](#)
- src/DotOutputVisitor.cpp, [48](#)
- src/DotOutputVisitor.hh, [48](#)
- src/InterfaceNodeVisitor.hpp, [50](#)
- src/MarkupNode.cpp, [50](#)
- src/MarkupNode.hh, [51](#)
- src/Node.cpp, [53](#)
- src/Node.hh, [53](#)
- src/OutputVisitor.cpp, [54](#)
- src/OutputVisitor.hh, [55](#)
- src/TextNode.cpp, [56](#)
- src/TextNode.hh, [57](#)
- TextNode
  - xml : :InterfaceNodeVisitor, [24](#)
  - xml : :TextNode, [42](#)
- textNodesCount
  - xml : :DotOutputVisitor, [21](#)
- visit
  - xml : :DotOutputVisitor, [20](#)
  - xml : :InterfaceNodeVisitor, [23](#), [24](#)
  - xml : :OutputVisitor, [37](#)
- writeAttributes
  - xml : :DotOutputVisitor, [20](#)
- writeDot
  - xml : :DotOutputVisitor, [20](#)
- writeln
  - xml : :DotOutputVisitor, [20](#)
- writeMarkupOpening
  - xml : :OutputVisitor, [37](#)
- xml, [9](#)
- xml : :CompositeMarkupNode, [11](#)
  - ~CompositeMarkupNode, [15](#)
  - \_Children, [14](#)
  - \_children, [16](#)
  - \_proxyToSelf, [16](#)
  - accept, [15](#)
  - begin, [15](#)
  - Children, [14](#)
  - ChildrenIterator, [14](#)
  - CompositeMarkupNode, [15](#)
  - end, [15](#)
- xml : :DotOutputVisitor, [16](#)
  - ~DotOutputVisitor, [19](#)
  - \_document, [20](#)
  - \_nodesCounter, [20](#)
  - \_out, [21](#)
  - declareNode, [20](#)
  - DotOutputVisitor, [19](#)
  - GRAPH\_TYPE, [21](#)
  - LINK, [21](#)
  - textNodesCount, [21](#)
  - visit, [20](#)
  - writeAttributes, [20](#)
  - writeDot, [20](#)

writeld, [20](#)

xml : :InterfaceNodeVisitor, [21](#)

    ~InterfaceNodeVisitor, [23](#)

    CompositeMarkupNode, [24](#)

    MarkupNode, [24](#)

    TextNode, [24](#)

    visit, [23](#), [24](#)

xml : :MarkupNode, [24](#)

    ~MarkupNode, [28](#)

    \_Attributes, [27](#)

    \_attributes, [29](#)

    \_name, [29](#)

    \_namespace, [29](#)

    accept, [28](#)

    Attributes, [27](#)

    AttributesIterator, [27](#)

    begin, [28](#)

    end, [28](#)

    MarkupNode, [28](#)

    name, [28](#)

    ns, [28](#)

xml : :Node, [29](#)

    ~Node, [32](#)

    \_parent, [32](#)

    accept, [32](#)

    Node, [32](#)

    parent, [32](#)

xml : :OutputVisitor, [33](#)

    ~OutputVisitor, [36](#)

    \_indent, [38](#)

    \_indentChar, [38](#)

    \_indentUnit, [38](#)

    \_out, [38](#)

    ASSIGN\_ATTRIBUTE\_VALUE\_STR, [38](#)

    CLOSE\_ATTRIBUTE\_VALUE\_STR, [38](#)

    CLOSE\_MARKUP\_STR, [38](#)

    CLOSING\_MARKUP\_STR, [38](#)

    DEFAULT\_INDENT\_CHAR, [38](#)

    DEFAULT\_INDENT\_UNIT, [38](#)

    doIndent, [37](#)

    INSIDE\_MARKUP\_SPACE\_STR, [39](#)

    NS\_SEPARATOR\_STR, [39](#)

    OPEN\_ATTRIBUTE\_VALUE\_STR, [39](#)

    OPEN\_MARKUP\_STR, [39](#)

    OutputVisitor, [36](#)

    visit, [37](#)

    writeMarkupOpening, [37](#)

xml : :TextNode, [39](#)

    ~TextNode, [42](#)

    \_content, [43](#)

accept, [42](#)

content, [42](#)

TextNode, [42](#)